Terrain-Aware Autonomous Exploration of Unstructured Confined Spaces

Héctor Azpúrua, Mario F. M. Campos, and Douglas G. Macharet

Laboratório de Visão Computacional e Robótica (VeRLab)
Programa de Pós-Graduação em Ciência da Computação
Universidade Federal de Minas Gerais (UFMG)
Belo Horizonte, MG – Brazil

{hector.azpurua,mario,doug}@dcc.ufmg.br

Abstract. *This work addresses the problem of exploring confined environments autonomously using terrestrial mobile robots. We propose a methodology for path planning in rough three-dimensional terrains and an exploration strategy that uses the map's navigable areas, the related navigation cost, and the information expected from a frontier to select the next promising exploration sector. The safe path generation algorithm models the environment as a graph and uses a linear combination of weights applied to multiple traversability metrics of the terrain. The exploration phase uses the expected volumetric information of frontiers; this way, the exploration areas are selected according to their expected utility and visitation cost. We also propose an online planning phase using the raw point cloud to avoid obstacles. The online phase is faster to compute and uses the MI-RRT algorithm, an RRT-based planner biased towards the most informative frontier, capable of simultaneously planning and selecting a target frontier.*

Keywords*: Exploration, Confined and Subterranean Spaces, Terrain-aware path planning.*

Ph.D. dissertation defended in 08/07/2022 (PPGCC/UFMG). Committee: Prof. Douglas G. Macharet (DCC/UFMG), Prof. Mario F. M. Campos (DCC/UFMG), Prof. Luiz Chaimowicz (DCC/UFMG), Prof. Gustavo Medeiros Freitas (DEE/UFMG), Prof. Gustavo Pessin (MI/ITV), and Prof. Denis Fernando Wolf (ICMC/USP). Full text, publications and videos are available at: h3ct0r.github.io/phd

1. Introduction

Confined areas are common in many industries: routine inspections, exploration, and surveillance are tasks that could benefit from using autonomous mobile robots. However, despite the advances in the field, there are still challenges to overcome for robot operation in those situations. Some of the principal challenges ground robots face in subterranean or enclosed scenarios lie in the lack of external or global localization, communication interference, and complex and rugged terrain topography (Figure 1). These particular characteristics force ground autonomous exploring devices to use terrain-aware path planning methods that could generate safer paths and minimize risks during robot locomotion. These hard challenges are gaining attention from the robotic academy and industry: even the latest DARPA (SubT) Challenge (2021), a highly regarded robotic competition, focused in underground environments [Allen 2019].

Confined areas are not only challenging for robots, but they also present hazardous situations for humans, such as gases, risk of entanglement, among others, so using autonomous robots instead of human personnel improves operational safety. For complex

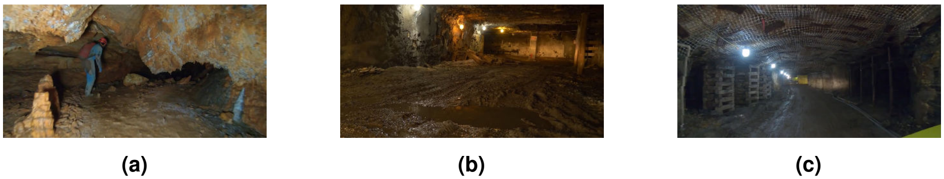


Figure 1. Examples of confined subterranean spaces: caves (a) and industrial tunnel systems (b-c), from the DARPA Subterranean Challenge.

scenarios, such as subterranean caves, traditional 2D navigation techniques cannot capture enough details of the environment to be successful for planning and exploration. Thereof, we present a complete pipeline for autonomous three-dimensional exploration tailored for subterranean confined spaces and environments alike. This work tackles the three-dimensional exploration problem using an information-theoretic approach that considers terrain traversability and expected mutual information gain as metrics for frontier selection. Unlike traditional techniques, we use multiple 3D map representations such as octrees, meshes, and point clouds to extract valuable data from the terrain to estimate safer, and less bumpy navigation paths; hence, the paths are generated by a weighted deterministic or probabilistic graph search algorithm combining multiple terrain metrics. In this sense, the proposed terrain-aware exploration algorithms consider if a frontier is reachable and the safest way to reach it. The frontier's information gain is estimated over the reconstructed scenario by measuring the expected information sensed by the robot. Also, unlike similar works, our proposed pipeline can successfully perform exploration missions with a limited sensorial kit of only one 3D LiDAR sensor, an IMU, and wheel encoders.

1.1. Contributions

The main contributions from this work are presented below, grouped according to the different challenges related to the problem.

1. **Path planning in rugged spaces**:
- We proposed a practical method for terrain-aware 3D navigation for terrestrial robots in confined and rugged spaces. In our proposal, the 3D point cloud generated by an online SLAM method is reconstructed into a mesh and modeled as a graph with terrain-aware edge weights. The optimal path uses a combination of metrics, optimized by a graph search algorithm.

2. **Exploration in confined and subterranean spaces**:
- **Static environment**: we proposed a new method for autonomous exploration that leverages the previous navigation scheme to generate efficient traversable paths that consider static obstacles and narrow passages. We used the 3D sensor parameters to project the expected best view of a frontier and estimate all reachable frontiers' Information Gain. The next best frontier selected for visitation uses a tradeoff between the navigation cost and the estimated frontier information gain.
- **Dynamic environment**: we present a methodology for autonomous 3D exploration using an efficient probabilistic framework capable of online obstacle avoidance. The method uses a biased RRT algorithm for global planning (MI-RRT) and a traditional RRT for local planning. Both sampling-based algorithms are faster by working directly with the raw point cloud. Furthermore, MI-RRT, uses the expected information gain of the frontiers as biases for expanding the RRT tree, thus reducing the time spent in an exploration cycle by performing the frontier selection and path planning at the same step.

The direct contributions of this work were published in the principal conferences in the area [Azpúrua et al. 2021c, Azpúrua et al. 2021b, Azpúrua et al. 2021a]. We also generated a comprehensive survey of the state-of-the-art in the autonomous exploration of confined spaces that is currently in revision [Azpúrua et al. 2022], and selected to participate in the RSS Pioneers workshop [Azpúrua et al. 2021a].

2. Navigation in rugged and confined spaces

Natural caves commonly present rough terrain, which is challenging for path planning. Complex landscapes require a complete three-dimensional map for planning a safe and efficient robotic locomotion. In this sense, three-dimensional meshes are adequate for this task since they can represent any 3D shape, generating highly descriptive environment models. Unlike 3D point clouds, meshes have the face area, position, and normal information, which help terrestrial robot mobility. Other representations such as elevation maps have difficulties representing some cave structures, e.g., arcs, narrow tunnels, or terrain overhangs. In that regard, we extend the path planning pipeline presented in [Santos et al. 2018] and [Azpurua et al. 2019] by integrating an automatic mesh generation process inside the planner, including a pre-processing step for filtering non-reachable regions and a fast method to compute the interaction of the robot with the ground. In contrast to our previous works in path planning, this pipeline is fast enough to be executed online at the exploration and mapping phases directly on the robot's onboard computer. The proposal is also fully integrated with a low-level control algorithm based on Vector Fields. Figure 2 shows the proposed path planning pipeline.

For the LiDAR-SLAM, we use the Lightweight and Ground-Optimized Lidar Odometry And Mapping (LeGO-LOAM) methodology proposed by [Shan and Englot 2018], which is a light-weight version of the LOAM technique, optimized for land vehicles since it assumes there is always a ground plane in the scan as captured via a multi-line LiDAR sensor. This method was validated for indoor, closed spaces and underground caves, thus presenting itself as a practical mapping method for confined spaces.

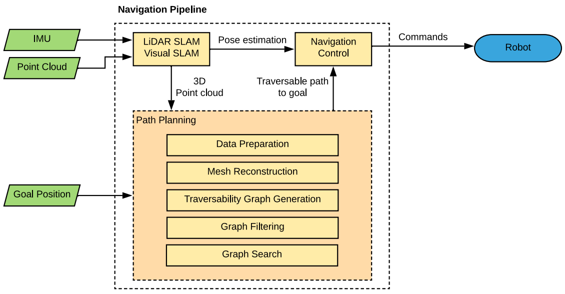


Figure 2. General path planning and navigation workflow.

2.1. Mesh Reconstruction

Mesh objects are generally a well-behaved set of joined faces without empty spaces between vertexes, in contrast to traditional point clouds. Therefore, in this work, we need to convert a point cloud into a coherent mesh object to estimate terrain-aware paths. Mesh reconstruction is generally an error-prone process that needs considerable parameter tuning and consumes a significant amount of CPU resources. Automatic reconstruction of cave-like environments is a particular challenge for reconstruction algorithms since the algorithm must preserve holes and other delicate components of the environment to generate a faithful three-dimensional mesh model. Noise and misalignments are common in point clouds generated by mobile robots; therefore, a reconstruction method must tolerate these types of errors. In this work we adapted the *Surface Recon*[1] mesh reconstruction algorithm proposed by Potje *et al.*, which provides an intuitive set of parameters available for non-expert users to adjust and, in most cases, works out-of-the-box. These characteristics made *Surface Recon* an adequate proposal for an online and automated mesh generation pipeline. The adapted algorithm has three main steps: (i) Normal estimation, (ii) Surface reconstruction, and (iii) hole filling. The color embedding methods from the original algorithm were removed from the pipeline to speed up the process since we are dealing with LiDAR data without any texture attached.

At the final steps, the holes are filled in the trimmed mesh using the technique presented by [Liepa 2003], with a threshold based on the average point distance to prevent cave entrances and other passages from being filled. The sequential process of mesh generation is depicted in Figure 3.

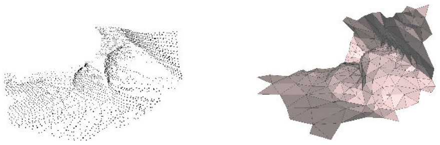


Figure 3. Example process for converting a raw point cloud of the environment into a mesh: (left) input point cloud, and (right) final mesh.

2.2. Path planning

Estimating reachable and non-reachable regions is particularly crucial for ground robots performing exploration tasks autonomously in uneven terrains. To generate a traversability map, we use the reconstructed mesh of the environment $\mathbb{M}$ in order to estimate slopes. The slopes are estimated by extracting the z normal vector ($\vec{\mathbf{n}}^z$) for every face of the mesh.

The mesh has a collection of faces $\mathbb{F}$, and it is filtered considering the maximum slope angle traversable by the robot (θ_{max}). We generate a graph $\mathcal{G} = (V, E)$, where the vertices V are the centroids of the mesh faces, and the edges E connect neighboring faces. The remaining unconnected traversable stages (such as stairs or similar structures) can be connected via a *bumpiness* threshold, calculated from a 3D sphere with a fixed radius from the closest vertices between the sub-graphs using a KD-tree search. Finally, we inflate borders and obstacles on $\mathcal{G}$ to prevent paths from being too close to dangerous areas. The complete process is summarized in Algorithm 1.

2.3. Path cost estimation

The paths to reach the frontiers are determined by using the well-known Dijkstra shortest path algorithm over the traversability graph. We define a path $p = \{n_1, ..., n_{goal}\}$ as an ordered subset of neighboring nodes from $\mathcal{G}$ and $\bar{p} = p \backslash n_1$, where n_1 is the robot current pose $\mathbf{q}$ and n_{goal} is the location of a frontier $f \in \mathcal{F}$. We use a metric that performs a

[1] www.github.com/verlab/mesh-vr-reconstruction-and-view

Algorithm 1: Traversability graph estimation

```
$\mathbb{M} \leftarrow$ generateMesh($\mathcal{M}$)
for $i \leftarrow 1$ to $|\mathbb{F}|$ do
    if $\vec{\mathbf{n}}_i^z < \theta_{max}$ then
        $\hat{\mathbb{M}} \leftarrow \mathbb{M} \setminus \{\mathbb{F}_i\}$                ▷ Remove face
$\mathcal{G} \leftarrow$ graphFromFaceCentroids($\hat{\mathbb{M}}$)
$\mathcal{G} \leftarrow$ KDTreeTraversablePlatformConnect($\mathcal{G}$)
$\mathcal{G} \leftarrow$ removeNonConnectedComponents($\mathcal{G}$)
$\mathcal{G} \leftarrow$ inflateBorders($\mathcal{G}$)
return $\mathcal{G}$
```

linear combination of the normalized Euclidean distance, terrain traversability, and energy consumption, as proposed in [Azpúrua et al. 2021c]. The final path cost is given by:

$$C(\tau) = \sum_{n \in \tau} \left[0.25 * N_d D(n) + 0.5 * N_t T(n) + 0.25 * N_e E(n)\right], \quad (1)$$

where N_d, N_t e N_e are normalization coefficients. The Euclidean distance $D(n)$ is the cumulative sum of the 3D Euclidean distance for neighboring nodes in the path and the traversability metric is based on the positive angle $T(n)$ between the normal $\vec{\mathbf{n}}_i^z$ of face i and the Z-axis ($\vec{Z}$):

$$D(n) = \sum_{n \in \bar{p}} d(n, n-1), \text{ and } T(n) = \arccos\left(\frac{|\vec{\mathbf{n}}_i^z \cdot \vec{Z}|}{\|\vec{\mathbf{n}}_i^z\| \|\vec{Z}\|}\right). \quad (2)$$

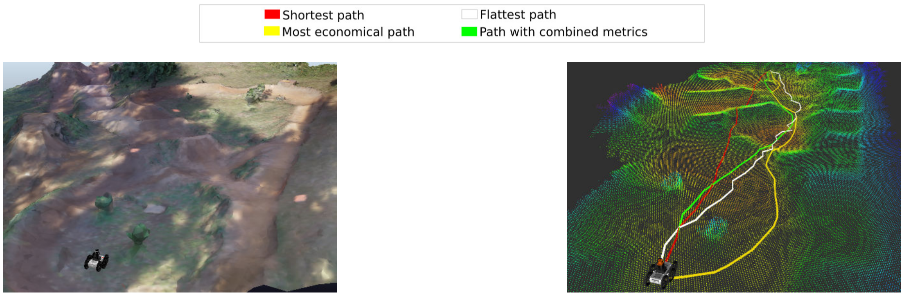
The energy consumption is estimated as a linear regression of the battery consumption, terrain inclination, friction coefficient, robot mass, and angle θ between the vectors linking neighboring mesh centers:

$$E(n) = \left(\frac{E_{mean}\ \alpha(n)}{2\pi}\right) + (a\theta(n) + b)D(n), \quad (3)$$

where E_{mean} is the mean battery consumption while turning 2π rad and a, b are the linear regression parameters. The quantities $D(n)$ and $\alpha(n)$ are estimates of the angular and linear displacements, respectively, when the robot moves between nodes n and $n + 1$. Finally, the robot navigates the path to visit a frontier using a vector field navigation algorithm with a potential-field based obstacle avoidance.

The proposed path planning and navigation pipeline was validated in virtual and real representative scenarios. Figure 4a shows an virtual scene obtained via photogrammetry reconstruction. Figure 4b shows the point cloud used to compute the cost functions of each metric. The red, white, and yellow paths correspond to the shortest, flattest, and most economical paths, respectively. The green path was obtained with the combined cost function. The shortest path tends to perform as a direct line to the goal, bypassing obstacles and other risk areas of the map. On the other hand, the most economical and flat paths tend to minimize terrain roughness and high slopes. The gains of the combined metric were defined as: $P_d = 0.5$, $P_t = 0.25$ and $P_e = 0.25$.

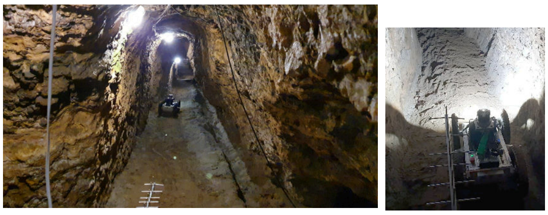
The real mine experiment was performed at the Mina du Veloso gold mine, located in Ouro Preto – MG, Brazil. Figure 5 shows the environment and robot setup. Mina du Veloso is a 400-year-old colonial gold mine with almost 300 meters of narrow and rugged multi-level corridors. The navigation and inspection experiment results inside the gold mine can be observed in Figure 6, where the robot performed mapping over ≈55 m of connected cave tunnels autonomously.



(a) Simulated environment in CoppeliaSim.

(b) Point cloud with estimated paths.

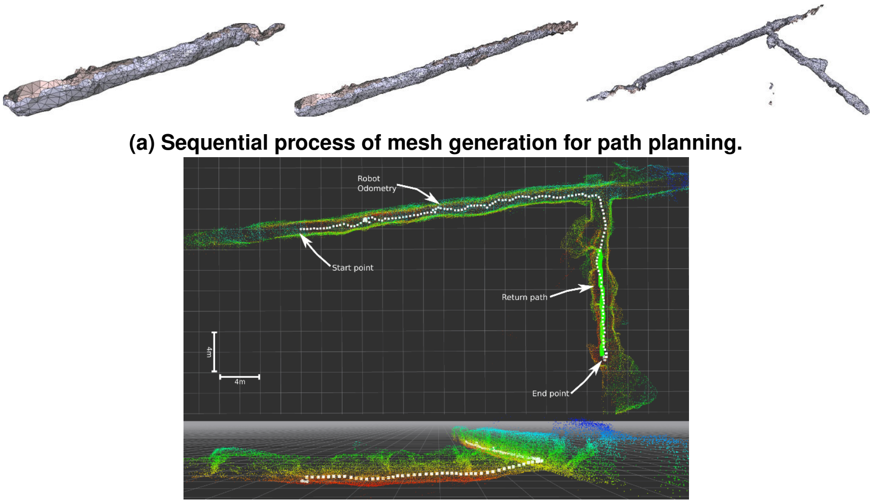
Figure 4. Path planning validation in simulated reconstruction of a rugged outdoor environment.



(a) Wide angle of the cave environment.

(b) Robot setup.

Figure 5. Experimental setup of the EspeleoRobô at the Mina du Veloso gold mine.



(a) Sequential process of mesh generation for path planning.

(b) Top and side view of the complete map, depicting one instance of path planing (green line). Odometry is signalized as white dots.

Figure 6. Inspection pipeline real experiment at the Mina du Veloso gold mine ($\approx 55m$). Top: (a) sequential coverage and the correspondings map meshes used to calculate the paths. Bottom: (b) final point cloud of the exploration pipeline (top and lateral view). The LiDAR odometry of the robot is depicted in a white dotted line, and the navigation path is shown in green.

3. Deterministic exploration (static environment)

We tackle the problem of an autonomous exploration with no initially prior information in confined environments, such as subterranean mines, tunnels, and caves. The task will be executed by a single ground robot R, where its pose q is represented by a configuration $\mathbf{q}_\mathrm{k} \in \mathrm{SE}(3)$. The robot must map a static environment $\mathcal{E} \in \mathbb{R}^3$, which poses critical challenges for the navigation, for example, obstacles, uneven terrain, and narrow passages. Let $\mathcal{M}$ be a three-dimensional occupancy grid representation of $\mathcal{E}$, generated by the observations of a 3D range sensor. The map will be initially set to $\mathcal{M} = \mathcal{E}_{unknown}$, as we do not assume any previous information on the environment. Space already explored ($\mathcal{E}_{known}$) can be either mapped into $\mathcal{M}_{free}$ (visited cells that do not contain any obstacle at the time of measurement) or $\mathcal{M}_{occupied}$ (cells with more than 0.5 probability of occupation given the sensor model). Given an initial position $n_i \in \mathcal{M}_{free}$, to reach a goal position n_{goal} we must define a safe and efficient continuous path $p = \{n_1,\ n_2,\ ...,\ n_n\} \to \mathcal{M}_{free}$, such that $p_0 = n_i$ and $p_n = n_{goal}$. Finally, a fundamental aspect of the exploration task is the selection of a location to visit. Therefore, based on the concept of frontier, given a collection of reachable frontiers $\mathcal{F}$, we must select the one that looks more promising to aggregate information to our map.

As proposed in the previous path planning section, a mesh reconstructed from the estimated occupational grid serves as input for generating a traversability graph with the robot's reachable regions. Nevertheless, in this case, we are also using Octrees to represent the environment's occupied and free areas. This representation of the environment is handy to perform ray casting and other topological analyses. In this regard, the information gain and the path cost are calculated for every extracted frontier using both an octree and a reconstructed mesh.

3.1. Reachable frontiers extraction

The traversability graph $\mathcal{G}$ and the complete mesh $\mathbb{M}$ are used to estimate the map frontiers. The known traversable areas that are neighbors of unexplored regions, including regions edging the map's borders without being an obstacle, are considered frontiers. Therefore a face in the mesh is considered as a frontier if:

$$\text{isFrontier}(\mathbb{M}, i) = \begin{cases} 1, & \text{if } |N_{\mathbb{F}}(i)| \leq 2 \\ 0, & \text{otherwise,} \end{cases} \tag{4}$$

where $N_{\mathbb{F}}(i)$ is the set of neighboring faces of face i.

The extracted frontiers that do not belong to the traversability graph are removed. This way, unreachable frontiers are eliminated from the pipeline. The frontier points are clustered into groups by their Euclidean distance using the Density-based spatial clustering algorithm (DBSCAN) [Schubert et al. 2017], considering a distance (eps) and minimum group size ($\mathcal{C}_{min}$). Finally, we use the KD-tree search to determine the visit location of a frontier cluster as the closest reachable point in $\mathcal{G}$ to the cluster's centroid. The frontier estimation process is described in Algorithm 2, where $\mathcal{B}$ are the raw mesh frontiers, $\mathcal{C}$ the clusters of reachable frontiers, and n a reachable node from G used as frontier visit point.

Algorithm 2: Reachable frontiers extraction

```
$\mathcal{B} \leftarrow \{\}, \mathcal{F} \leftarrow \{\}$
for $i \leftarrow 1$ to $|\mathbb{F}|$ do
    if isFrontier($\mathcal{M}, i$) then $\mathcal{B} \leftarrow \mathcal{B} \cup \mathbb{F}_i$          ▷ Raw mesh frontiers;
$\mathcal{B}' \leftarrow$ traversabilityFilter($V, \mathcal{B}$)
$\mathcal{C} \leftarrow$ DBSCAN($\mathcal{B}', eps, \mathcal{C}_{min}$)
for $c \in \mathcal{C}$ do
    $n \leftarrow$ getCentroidReachableNode($\mathcal{G}, c$)
    $\mathcal{F} \leftarrow \mathcal{F} \cup n$          ▷ Reachable frontiers centroids
return $\mathcal{F}$
```

3.2. Information-theoretic frontier selection

To estimate the utility of a frontier, we use the mutual information gain metric over the 3D map $\mathcal{M}$, and the cost of the path $C(\tau)$ to reach the frontier. The mutual information uses the probability of the octree cells to calculate the current entropy of the map and then compares it with the expected entropy of the map after performing a virtual exploration at the selected frontier. The entropy is defined as:

$$H(\mathcal{M}) = -\sum_{i,j,k} p_{ijk}\ \log(p_{ijk}), \tag{5}$$

where $\mathcal{M}$ is the current map, and p_{ijk} is the outcome of the Bernoulli random variable representing the cell occupation. The mutual information $I(\mathcal{M}, f_i)$ is used to obtain the expected information gain by visiting frontier f_i, i.e.:

$$I(\mathcal{M}, f_i) = H(\mathcal{M}) - H(\mathcal{M}|f_i), \tag{6}$$

where $H(\mathcal{M}|f_i)$ is the expected new entropy. The *virtual exploration* phase uses the sensor model maximum range and field of view to project rays in the current map $\mathcal{M}$ considering the vehicle will be in the centroid of each frontier in $\mathcal{F}$. Given the current map $\mathcal{M}$ state, the projected rays give a reasonable estimate of the maximum free volume the robot could sense of a frontier. Since the measurements from these types of multi-line LiDAR sensors results in a sparse Octree, we performed a filling step at the virtual map using a uniform sampling of the mesh surfaces $\mathbb{F}$, preventing the projected rays from escaping through walls and solid objects over the occupancy grid $\mathcal{M}$. Finally, the metric for selecting the best frontier is:

$$c^{*} = \underset{\forall f \in \mathcal{F}}{\arg\max} \frac{MI(\mathcal{M}, f) + e}{C(p^{f})}, \tag{7}$$

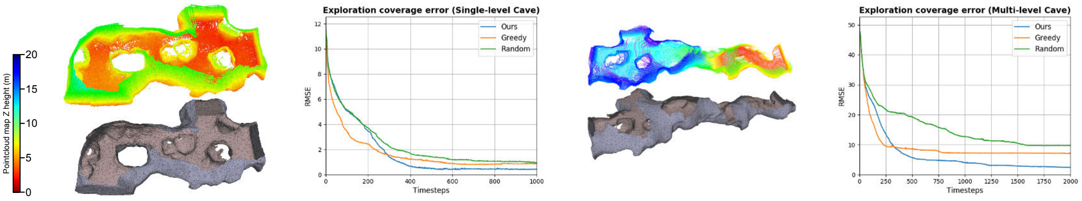
where p^{f} is a path from the robot's current position to frontier f, and e is a small tolerance constant.

The deterministic exploration pipeline was validated in representative scenarios from the DARPA SubT challenge[2]. The resulting maps and the reconstructed meshes for the simulated scenarios can be observed in Fig. 7. We validated multiple frontier and path selection metrics: (i) our proposed approach using the information gain and traversability path generation, (ii) a greedy selection of the closest next frontier using the path that gives the smallest Euclidean distance instead of the terrain aware one, and (iii) random selection of the frontier using a terrain aware path. The results of this analysis can be observed at the right side of Figs. 7a and 7b, where the displayed lines are the mean RMSE of the map coverage for every timestep (10 repetitions). The proposed metric (blue) converges to a lower error rate than the other metrics at all environments, even if the terrain-friendly paths generated by the proposed metric are longer than the shortest euclidean path.

4. Probabilistic exploration (dynamic environment)

Different from the deterministic exploration, which considers mapping a static environment, here we also consider additive obstacles in $\mathcal{E}$, meaning that the environment is not static anymore. We consider an environment dynamic if there is a chance of the robot encountering an additive environmental modification. An additive obstacle or additive

[2] Deterministic exploration video:`https://youtu.be/RrkOvAZEteQ`



(a) Single-level cave scenario

(b) Multi-level cave scenario

Figure 7. Exploration analysis. Estimated point-cloud (colored) and final reconstructed mesh (brownish) for the cave environments: (a) a single-level cave, (b) a multi-level cave map.

modification of the environment is any addition of points in the map related to a new object that was not initially there. Examples of additive obstacles are collapsed structures, closed doors (previously open), or other elements that could suddenly appear in $\mathcal{E}$. As before, given a collection of reachable frontiers $\mathcal{F}$, we must select the one that looks more promising, both in navigation easiness and expected information gain.

Unlike the previous approach, this proposal uses the raw point clouds for planning using a RRT-biased planner (MI-RRT) towards the more informative regions. The use of point clouds instead of the complete reconstruction pipeline for planning improves the efficiency of the terrain analysis at the trade-off of a decreased connectivity precision when generating the traversability graph; however, as seen in the experiments, the difference in exploration performance is slight while the computing time is significantly decreased. The proposed method is divided into three big steps: (i) frontier extraction, (ii) global path planning, and (iii) local path planning. The frontier extraction procedure (i) uses the same steps as before, extracting a traversability graph from a mesh of the environment, which we use to cluster the reachable frontiers and estimate the information gain of them.

4.1. MI-RRT global planner with information bias

The global planner complements the sparse point cloud estimated in the SLAM step with a denser cloud extracted from the mesh generated in the frontier extraction process. Sampling the mesh has several advantages over only using the SLAM point cloud directly, allowing for a denser representation, thus smoothing the generated final paths and filling small holes in the map. The joint point cloud $\hat{\mathcal{M}}$ is then prepared in several steps to guarantee traversability. The cloud is first voxelized, and then the normals of the points are estimated. Next, the points with a normal greater than the maximum slope angle traversable by the robot (θ_{max}) are filtered. The adjacent points of those defined as not traversable ($\mathbb{X}$) are removed as well by using a search by radius ($\tau_{inflation}$). The points closer to the border are detected using the KD-Tree algorithm. After removing the border points, a clustering algorithm (DBSCAN) is used to group the points closer to the initial location of the robot. The final point cloud $\hat{\mathcal{M}}$ then possesses only traversable points, without obstacles and with a safe margin between traversable regions and the obstacles. The traversable graph $\mathcal{G}$ is then generated by a k-NN algorithm with a Euclidean distance filter (radius τ_{bump}).

Previously, we used the exact metric described in Equation 7 to select the best frontier among the frontier's set. Here, we propose to adapt the two-dimensional exploration method presented in [Pimentel et al. 2018] into a fully 3D environment, where a customized RRT expands to the frontiers given the proportional information gain. This probabilistic method outperforms the previous deterministic frontier selection since it can perform the frontier selection directly at the path planning phase. The proposed RRT-based path search, called Mutual Information RRT (MI-RRT), is described in Algorithm 3. The key of the proposed approach is modifying the sampling part of the RRT planner to use biased point selection based on the information gain of frontiers (*sampleWithRoulette*($\mathcal{G}, V, \mathcal{F}'$)) function, where $\mathcal{F}'$ is the scaled set of information gain of the reachable frontiers in $\mathcal{G}$ in the $[0, 1]$ range. The roulette sampling uses the normalized information weight of the frontiers to select the next point to add to the tree. We also added an extra fixed probability of a completely random point to decrease possible local minima (τ_{random}). The final result of the global planner is a set of ordered vertices to be visited by the robot to reach a promising, informative frontier. This path has a low possibility of collisions and should be safe to traverse in a static environment. Nevertheless, here, the global path is used as a reference for the local planner.

Algorithm 3: MI-RRT path planning algorithm $(\mathcal{G}, \mathcal{N}, \mathcal{F}')$

```
$V \leftarrow \{x_{init}\}$
$E \leftarrow \emptyset$
for $i \leftarrow 1$ to $\tau_{iter}$ do
    $x_{biased} \leftarrow sampleWithRoulette(\mathcal{G}, V, \mathcal{F}')$
    $x_{nearest} \leftarrow nearest(\mathcal{G}, V, x_{biased})$
    $x_{new} \leftarrow steer(x_{nearest}, x_{biased})$
    if $obstacleFree(x_{nearest}, x_{new})$ then
        $V \leftarrow V \cup \{x_{new}\}$
        $E \leftarrow E \cup \{(x_{nearest}, x_{new})\}$
$\mathcal{G} = (V, E)$
$p_{global} \leftarrow extractPath(\mathcal{G})$
return $p_{global}$
```

4.2. RRT local planner with viewpoint filtering

For local planning, we also use the RRT algorithm, but instead of the complete map of the environment, the RRT operates over a minor point cloud representing the local surroundings of the robot. This local point cloud extraction uses an extension of the method proposed in [Katz et al. 2007], which approximates the visibility of a point cloud from a given view (in our case, robot pose) without surface reconstruction or normal estimation. The method efficiently removes points that the robot cannot view in a given location, such as occluded by walls, obstacles, or other robots. The path navigation is performed by sampling the global path p_{global} in a lookahead fashion – similar to the way a pure pursuit algorithm estimates the new point to visit. This way, the robot is chasing a “constantly moving point” over the global path that is at some distance ahead of it. The RRT planner then constantly estimates the path from the current robot position to the closest location to the sampled point[3]. Using a local point cloud is critical for planning safe paths with reachable regions and detecting when the global path cannot be followed anymore. A typical example of this is when the global path is interrupted by an obstacle. In this case, if the distance of the robot to the end of the local path is less than a threshold τ_{target}, we verify if the global target was reached and, if true, end this exploration cycle. On the other hand, if the threshold was reached but the global target is still far away, we assume that the robot ended the local path without reaching the frontier and needs to replan to another frontier.

[3] Local planner navigation video: `https://youtu.be/FN2rdfWwqpE`

The RRT planner was validated in realistic cave environments[4] against the best and worst frontier selections of the previous exact exploration method: (i) the proposed approach using the information gain and traversability path generation of the frontiers to select the next visit point, and (ii) the random selection of the frontier using the exact terrain-aware path. The results of this analysis can be observed in Fig. 8, where the displayed lines are the mean RMSE for every timestep of five repetitions per experiment. The standard deviation (σ) in the RRT metric is also shown in light purple. The RRT metric (purple) has a better performance than the orange metric (exact path planning with random frontier selection). However, the exact metric for frontier selection (MI ratio) still performs better than the others. The probabilistic nature of the RRT with bias could explain the inferior performance of the exact metric since there is no guarantee that the best path is selected at all times.

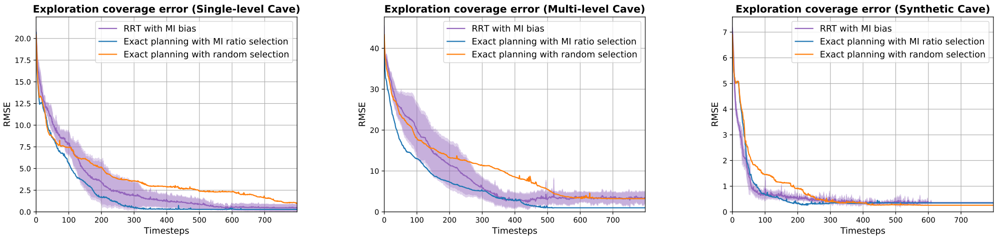


Figure 8. RRT exploration error (RMSE) mean of five runs comparing the real-time SLAM point cloud with a reference map for the: (a) single-level cave, (b) multi-level cave, and (c) synthetic cave.

5. Conclusions and Future work

This work has presented a pipeline for three-dimensional path planning for ground robots in rugged terrains and two incremental proposals for terrain-aware autonomous exploration for confined spaces. We proposed a deterministic exploration method using optimal path search algorithms and a probabilistic method using a biased RRT that we call MI-RRT. The basis of our exploration proposals uses a novel approach that combines the cost of traversing rugged terrains and the expected information gathered by visiting a frontier. Our proposed method uses octrees, meshes, and the raw point cloud to calculate the most informative frontiers and generate safe paths considering terrain traversability, distance, and energy consumption. Furthermore, unlike traditional exploration methods, the proposed method works in complex 3D environments without assuming any priors over the map structure. In contrast to other start-of-the-art works on real-world exploration in confined scenarios, in this work, we proposed using a limited sensorial suite composed of only one 3D LiDAR, an IMU, and wheel encoders. This limited suite was shown to be sufficient for terrain-aware exploration, being more economical, lightweight, and consuming less energy than other approaches, with few drawbacks. For example, the lack of a dense sensor, such as a projected depth camera, makes detecting small untraversable areas at the proximities of the robot a challenge since our 3D sensor is sparse. Of course, this is a concern for an actual robot deployment; however, in our tests, we did not observe dangerous situations generated by this condition.

The proposed exploration pipeline leads to numerous open questions and research directions, especially when dealing with multi-robot or learn-based approaches. For

[4] Probabilistic exploration video: `https://youtu.be/Otboyqh6GWI`

example, multiple robots could improve the efficiency and resiliency of exploratory robotic behaviors. In this regard, homogeneous terrestrial cooperation looks promising, where a group of centralized or decentralized robots performs the exploration without repeating paths between them, generating a comprehensive shared map combining the group's gathered information. However, cooperation is a challenge by itself, and it will need the research of task allocation methods appropriate for confined scenarios using robots with limited sensing and communication range to enable efficient multi-robot exploration behaviors.

References

Allen, B. (2019). Unearthing the subterranean environment.

Azpúrua, H., Campos, M. F., and Macharet, D. G. (2021a). Research statement: Towards terrain-aware autonomous exploration in 3d confined spaces. In *Robotics: Science and Systems 2021 (RSS)*. RSS. (Presented in the RSS Pioneers Workshop).

Azpúrua, H., Campos, M. F., and Macharet, D. G. (2021b). Three-dimensional terrain aware autonomous exploration for subterranean and confined spaces. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2443–2449. IEEE.

Azpúrua, H., Rezende, A., Potje, G., da Cruz Júnior, G. P., Fernandes, R., Miranda, V., de Resende Filho, L. W., Domingues, J., Rocha, F., de Sousa, F. L. M., et al. (2021c). Towards semi-autonomous robotic inspection and mapping in confined spaces with the espeleorobô. *Journal of Intelligent & Robotic Systems*, 101(4):1–27.

Azpurua, H., Rocha, F., Garcia, G., Santos, A. S., Cota, E., Barros, L. G., Thiago, A. S., Pessin, G., and Freitas, G. M. (2019). Espeleorobô-a robotic device to inspect confined environments. In *2019 19th International Conference on Advanced Robotics (ICAR)*, pages 17–23. IEEE.

Azpúrua, H., M. Freitas, G., Clark, L., Agha-mohammadi, A.-a., Pessin, G., F. M. Campos, M., and G. Macharet, D. (2022). A survey on the autonomous exploration of confined subterranean spaces: Perspectives from real-word and industrial robotic deployments. *Journal of Robotics and Autonomous Systems*. (In revision).

Katz, S., Tal, A., and Basri, R. (2007). Direct visibility of point sets. In *ACM SIGGRAPH 2007 papers*, pages 24–es. ACM.

Liepa, P. (2003). Filling holes in meshes. In *Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 200–205.

Pimentel, J. M., Alvim, M. S., Campos, M. F., and Macharet, D. G. (2018). Information-driven rapidly-exploring random tree for efficient environment exploration. *Journal of Intelligent & Robotic Systems*, 91(2):313–331.

Santos, A. S., Azpúrua, H. I. P., Pessin, G., and Freitas, G. M. (2018). Path planning for mobile robots on rough terrain. In *LARS 2018*, pages 265–270. IEEE.

Schubert, E., Sander, J., Ester, M., Kriegel, H. P., and Xu, X. (2017). DBSCAN revisited, revisited: why and how you should (still) use DBSCAN. *ACM Transactions on Database Systems (TODS)*, 42(3):1–21.

Shan, T. and Englot, B. (2018). Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4758–4765. IEEE.